

1. In a network environment that includes at least a first and second computing system that are capable of communicating messages with each other in a message exchange pattern, a method for a first computing system enforcing the message exchange pattern to reduce the chance that messages that are invalid at any given point in the message exchange pattern are transmitted to the second computing system to thereby preserve network bandwidth and processing resources, the method comprising the following:

an act of tracking progress through the message exchange pattern;

an act of detecting a request from a component to transmit a message to the second computing system;

an act of determining that the message is not a valid message given the tracked progress through the message exchange pattern; and

an act of notifying the component that the message is not a valid message.

2. A method in accordance with Claim 1, wherein the act of tracking progress through the message exchange pattern comprises the following:

an act of maintaining a state transition tree representing the message exchange pattern, wherein transitions from a first state to a second state occur upon the transmission or receipt of one or more valid messages for the first state.

3. A method in accordance with Claim 1, wherein the message is a HyperText Transport Protocol (HTTP) message.

4. A method in accordance with Claim 1, wherein the message is a Simple Object Access Protocol (SOAP) message.

5. A method in accordance with Claim 4, wherein a type of message is specified in a SOAP header of the message, wherein the act of determining that the message is not a valid message given the tracked progress through the message exchange pattern comprises the following:

an act of reading the SOAP header of the message.

6. A method in accordance with Claim 1, wherein the message is an RMI invocation.

7. A method in accordance with Claim 1, wherein the message is a first message, the method further comprising the following:

an act of detecting a request from a component to transmit a second message to the second computing system;

an act of determining that the second message is a valid message given the tracked progress through the message exchange pattern; and

an act of transmitting the second message to the second computing system.

8. A method in accordance with Claim 7, further comprising the following:

an act of identifying a role of the first computing system in the message exchange pattern, wherein the act of determining that the second message is a valid message is performed in light of the identified role.

9. A method in accordance with Claim 7, further comprising the following:

an act of loading state information related to the message exchange pattern from persistent memory to system memory in response to the act of detecting a request to transmit the second message to the second computing system;

an act updating the state information to represent the transmission of the second message to the second computing system upon transmitting the second message to the second computing system; and

saving the updated state information.

10. A method in accordance with Claim 1, further comprising the following:

an act of loading state information related to the message exchange pattern from persistent memory to system memory in response to the act of detecting a request to transmit the first message to the second computing system.

11. A method in accordance with Claim 10, further comprising the following:

an act of clearing system memory of the updated state information upon the act of notifying the component that the message is not a valid message.

12. A method in accordance with Claim 1, wherein the message exchange pattern includes a plurality of application layer messages.

13. A method in accordance with Claim 12, wherein the message exchange pattern includes a plurality of protocol layer messages.

14. A method in accordance with Claim 1, wherein the message exchange pattern includes a plurality of protocol layer messages.

15. A method in accordance with Claim 1, wherein the message exchange pattern includes the transmission of one or more messages in which the message exchange pattern is identified and agreed to between the first computing system and the second computing system.

16. A method in accordance with Claim 1, further comprising the following:
an act of identifying a role of the first computing system in the message exchange pattern, wherein the act of determining that the message is not a valid message is performed in light of the identified role.

17. A computer program product for use in a network environment that includes at least a first and second computing system that are capable of communicating messages with each other in a message exchange pattern, the computer program product for performing a method for a first computing system enforcing the message exchange pattern to reduce the chance that messages that are invalid at any given point in the message exchange pattern are transmitted to the second computing system to thereby preserve network bandwidth and processing resources, the computer program product comprising one or more computer-readable media having thereon computer-executable instructions comprising that, when executed by one or more processors of the first computing system, cause the first computing system to perform the following:

- an act of tracking progress through the message exchange pattern;
- an act of detecting a request from a component to transmit a message to the second computing system;
- an act of determining that the message is not a valid message given the tracked progress through the message exchange pattern; and
- an act of notifying the component that the message is not a valid message.

18. A computer program product in accordance with Claim 17, wherein the one or more computer-readable media are physical media.

19. A computer program product in accordance with Claim 18, wherein the one or more computer-readable media includes system memory.

20. A computer program product in accordance with Claim 18, wherein the one or more computer-readable media includes persistent memory.

21. A computer program product in accordance with Claim 20, wherein the persistent memory is a magnetic disk.

22. A computer program product in accordance with Claim 17, wherein the act of tracking progress through the message exchange pattern comprises the following:

an act of maintaining a state transition tree representing the message exchange pattern, wherein transitions from a first state to a second state occur upon the transmission or receipt of one or more valid messages for the first state.

23. A computer program product in accordance with Claim 17, wherein the message is a HyperText Transport Protocol (HTTP) message.

24. A computer program product in accordance with Claim 17, wherein the message is a Simple Object Access Protocol (SOAP) message.

25. A computer program product in accordance with Claim 24, wherein a type of message is specified in a SOAP header of the message, wherein the act of determining that the message is not a valid message given the tracked progress through the message exchange pattern comprises the following:

an act of reading the SOAP header of the message.

26. A computer program product in accordance with Claim 17, wherein the message is an RMI invocation.

27. A computer program product in accordance with Claim 17, wherein the message is a first message, the method further comprising the following:

an act of detecting a request from a component to transmit a second message to the second computing system;

an act of determining that the second message is a valid message given the tracked progress through the message exchange pattern; and

an act of transmitting the second message to the second computing system.

28. A computer program product in accordance with Claim 27, wherein the one or more computer-readable media further have thereon computer-executable instructions that, when executed by the one or more processors, cause the first computing system to perform the following:

an act of identifying a role of the first computing system in the message exchange pattern, wherein the act of determining that the second message is a valid message is performed in light of the identified role.

29. A computer program product in accordance with Claim 27, further comprising the following:

an act of loading state information related to the message exchange pattern from persistent memory to system memory in response to the act of detecting a request to transmit the second message to the second computing system;

an act updating the state information to represent the transmission of the second message to the second computing system upon transmitting the second message to the second computing system; and

saving the updated state information.

30. A computer program product in accordance with Claim 17, further comprising the following:

an act of loading state information related to the message exchange pattern from persistent memory to system memory in response to the act of detecting a request to transmit the first message to the second computing system.

31. A computer program product in accordance with Claim 30, further comprising the following:

an act of clearing system memory of the updated state information upon the act of notifying the component that the message is not a valid message.

32. A computer program product in accordance with Claim 17, wherein the one or more computer-readable media further have thereon computer-executable instructions that, when executed by the one or more processors, cause the first computing system to perform the following:

an act of identifying a role of the first computing system in the message exchange pattern, wherein the act of determining that the message is not a valid message is performed in light of the identified role.

33. In a network environment that includes at least a first and second computing system that are capable of communicating messages with each other in a message exchange pattern, a method for a first computing system enforcing the message exchange pattern to reduce the chance that messages that are invalid at any given point in the message exchange pattern are transmitted to the second computing system to thereby preserve network bandwidth and processing resources, the method comprising the following:

an act of tracking progress through the message exchange pattern; and
a step for transmitting messages only if appropriate given the current tracked progress through the message exchange pattern.

34. A method in accordance with Claim 33, wherein the step for transmitting messages only if appropriate given the current tracked progress through the message exchange pattern comprises the following:

an act of detecting a request from a component to transmit a message to the second computing system;
an act of determining that the message is not a valid message given the tracked progress through the message exchange pattern; and
an act of notifying the component that the message is not a valid message.

35. A method in accordance with Claim 34, wherein the message is a first message, the step for transmitting messages only if appropriate given the current tracked progress through the message exchange pattern further comprising the following

an act of detecting a request from a component to transmit a second message to the second computing system;

an act of determining that the second message is a valid message given the tracked progress through the message exchange pattern; and

an act of transmitting the second message to the second computing system.

36. A method in accordance with Claim 34, wherein the step for transmitting messages only if appropriate given the current tracked progress through the message exchange pattern comprises the following:

an act of detecting a request from a component to transmit a message to the second computing system;

an act of determining that the message is a valid message given the tracked progress through the message exchange pattern; and

an act of transmitting the message to the second computing system.